

## Java programming 2\_3

1. Update the JavaBank.java application to use the toString() methods to display the bank account details to the text area in the Java application. a. Update the myAccounts array definition to use the AbstractBankAccount class as its base class. b. Update the displayAccountDetails() method to accept a single parameter of type AbstractBankAccount named account. c. Call the account objects toString() method to provide the text for the JTextArea. d. Update the method calls to displayAccountDetails() to pass a single account object as an argument. Ensure that all displays are carried out through the displayAccountDetails() method.

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class Bank {
    // AbstractBankAccount class
    public abstract static class AbstractBankAccount {
        protected String accountNumber;
        protected double balance;
        public AbstractBankAccount(String accountNumber, double balance) {
            this.accountNumber = accountNumber;
            this.balance = balance;
        }
        public String getAccountNumber() {
            return accountNumber;
        }
        public double getBalance() {
            return balance;
        }
    }
    @Override
    public String toString() {
        return "Account Number: " + accountNumber + "\nBalance: " + balance;
    }
}
// SavingsAccount class
public static class SavingsAccount extends AbstractBankAccount {
    private double interestRate;
    public SavingsAccount(String accountNumber, double balance, double interestRate) {
        super(accountNumber, balance);
        this.interestRate = interestRate;
    }
    @Override
    public String toString() {
        return super.toString() + "\nInterest Rate: " + interestRate;
    }
}
```

```

}
// CheckingAccount class
public static class CheckingAccount extends AbstractBankAccount {
    private double overdraftLimit;
    public CheckingAccount(String accountNumber, double balance, double overdraftLimit) {
        super(accountNumber, balance);
        this.overdraftLimit = overdraftLimit;
    }
    @Override
    public String toString() {
        return super.toString() + "\nOverdraft Limit: " + overdraftLimit;
    }
}
// JavaBank application

private AbstractBankAccount[] myAccounts;
private JTextArea displayArea;
public JavaBank() {
    myAccounts = new AbstractBankAccount[5]; // Example array size
    displayArea = new JTextArea(10, 30);
    // Example account initialization
    myAccounts[0] = new SavingsAccount("12345", 1000.0, 0.05);
    myAccounts[1] = new CheckingAccount("67890", 500.0, 100.0);
    JFrame frame = new JFrame("JavaBank");
    JButton displayButton = new JButton("Display Account Details");
    displayButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            displayAccountDetails(myAccounts[0]); // Display the first account details as an
            example
        }
    });
    frame.add(displayButton);
    frame.add(new JScrollPane(displayArea));
    frame.setLayout(new BorderLayout(frame.getContentPane(), BorderLayout.Y_AXIS));
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.pack();
    frame.setVisible(true);
}
private void displayAccountDetails(AbstractBankAccount account) {
    displayArea.setText(account.toString());
}
public static void main(String[] args) {
    new JavaBank();
}

```

```

}
2.package bikeproject;
public class BikeProject {
// BikeParts Interface
public interface BikeParts {
// Constant declaration
public final String MAKE = "Oracle Bikes";
// Required methods after implementation
public String getHandleBars();
public void setHandleBars(String newValue);
public String getTyres();
public void setTyres(String newValue);
public String getSeatType();
public void setSeatType(String newValue);
}
// MountainBike Class implementing BikeParts interface
public static class MountainBike implements BikeParts {
private String handleBars;
private String tyres;
private String seatType;
@Override
public String getHandleBars() {
return handleBars;
}

@Override
public void setHandleBars(String newValue) {
handleBars = newValue;
}
@Override
public String getTyres() {
return tyres;
}
@Override
public void setTyres(String newValue) {
tyres = newValue;
}
@Override
public String getSeatType() {
return seatType;
}
@Override
public void setSeatType(String newValue) {
seatType = newValue;
}
}

```

```

// Displaying the bike details
public void displayBikeDetails() {
    System.out.println("Bike Make: " + MAKE);
    System.out.println("HandleBars: " + getHandleBars());
    System.out.println("Tyres: " + getTyres());
    System.out.println("Seat Type: " + getSeatType());
}
}

// Main method to test the implementation
public static void main(String[] args) {
    MountainBike myBike = new MountainBike();
    myBike.setHandleBars("Drop");
    myBike.setTyres("Off-road");
    myBike.setSeatType("Comfort");
    myBike.displayBikeDetails();
}
}

3. package bikeproject;
public class MountainBike implements MountainParts {
    private String suspension;
    private String type;
    @Override
    public String getSuspension() {
        return suspension;
    }
    @Override
    public void setSuspension(String newValue) {
        suspension = newValue;
    }
    @Override
    public String getType() {
        return type;
    }
    @Override
    public void setType(String newValue) {
        type = newValue;
    }
}

// Displaying the mountain bike details
public void displayBikeDetails() {
    System.out.println("Terrain: " + TERRAIN);
    System.out.println("Suspension: " + getSuspension());
    System.out.println("Type: " + getType());
}
}

```

```

public static void main(String[] args) {
MountainBike myBike = new MountainBike();
myBike.setSuspension("Full");
myBike.setType("Trail");
myBike.displayBikeDetails();
}
}

```

Final program:

```

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class IntegratedProject {
// AbstractBankAccount class

public abstract static class AbstractBankAccount {
protected String accountNumber;
protected double balance;
public AbstractBankAccount(String accountNumber, double balance) {
this.accountNumber = accountNumber;
this.balance = balance;
}
public String getAccountNumber() {
return accountNumber;
}
public double getBalance() {
return balance;
}
@Override
public String toString() {
return "Account Number: " + accountNumber + "\nBalance: " + balance;
}
}
// SavingsAccount class
public static class SavingsAccount extends AbstractBankAccount {
private double interestRate;
public SavingsAccount(String accountNumber, double balance, double interestRate) {
super(accountNumber, balance);
this.interestRate = interestRate;
}

@Override
public String toString() {
return super.toString() + "\nInterest Rate: " + interestRate;
}
}
}

```

```

// CheckingAccount class
public static class CheckingAccount extends AbstractBankAccount {
    private double overdraftLimit;
    public CheckingAccount(String accountNumber, double balance, double overdraftLimit) {
        super(accountNumber, balance);
        this.overdraftLimit = overdraftLimit;
    }
    @Override
    public String toString() {
        return super.toString() + "\nOverdraft Limit: " + overdraftLimit;
    }
}

// JavaBank application
public static class JavaBank {
    private AbstractBankAccount[] myAccounts;
    private JTextArea displayArea;
    public JavaBank() {
        myAccounts = new AbstractBankAccount[5]; // Example array size
        displayArea = new JTextArea(10, 30);
        // Example account initialization

        myAccounts[0] = new SavingsAccount("12345", 1000.0, 0.05);
        myAccounts[1] = new CheckingAccount("67890", 500.0, 100.0);
        JFrame frame = new JFrame("JavaBank");
        JButton displayButton = new JButton("Display Account Details");
        displayButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                displayAccountDetails(myAccounts[0]); // Display the first account details as an example
            }
        });
        frame.add(displayButton);
        frame.add(new JScrollPane(displayArea));
        frame.setLayout(new BorderLayout(frame.getContentPane(), BorderLayout.Y_AXIS));
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
    private void displayAccountDetails(AbstractBankAccount account) {
        displayArea.setText(account.toString());
    }
}

// BikeParts Interface
public interface BikeParts {

```

```

// Constant declaration
public final String MAKE = "Oracle Bikes";

// Required methods after implementation
public String getHandleBars();
public void setHandleBars(String newValue);
public String getTyres();
public void setTyres(String newValue);
public String getSeatType();
public void setSeatType(String newValue);
public String getSuspension();
public void setSuspension(String newValue);
public String getType();
public void setType(String newValue);
}

// MountainBike Class implementing BikeParts interface
public static class MountainBike implements BikeParts {
    private String handleBars;
    private String tyres;
    private String seatType;
    private String suspension;
    private String type;
    @Override
    public String getHandleBars() {
        return handleBars;
    }

    @Override
    public void setHandleBars(String newValue) {
        handleBars = newValue;
    }
    @Override
    public String getTyres() {
        return tyres;
    }
    @Override
    public void setTyres(String newValue) {
        tyres = newValue;
    }
    @Override
    public String getSeatType() {
        return seatType;
    }
    @Override
    public void setSeatType(String newValue) {

```

```

seatType = newValue;
}
@Override
public String getSuspension() {
return suspension;
}
@Override

public void setSuspension(String newValue) {
suspension = newValue;
}
@Override
public String getType() {
return type;
}
@Override
public void setType(String newValue) {
type = newValue;
}
// Displaying the bike details
public void displayBikeDetails() {
System.out.println("Bike Make: " + MAKE);
System.out.println("HandleBars: " + getHandleBars());
System.out.println("Tyres: " + getTyres());
System.out.println("Seat Type: " + getSeatType());
System.out.println("Suspension: " + getSuspension());
System.out.println("Type: " + getType());
}
public static void main(String[] args) {
MountainBike myBike = new MountainBike();
myBike.setHandleBars("Drop");
myBike.setTyres("Off-road");
myBike.setSeatType("Comfort");
myBike.setSuspension("Full");
myBike.setType("Trail");

myBike.displayBikeDetails();
}
}
// Main method to run the entire project
public static void main(String[] args) {
// Start JavaBank application
// Test the MountainBike class
MountainBike myBike = new MountainBike();
myBike.setHandleBars("Drop");

```



```
myBike.setTyres("Off-road");  
myBike.setSeatType("Comfort");  
myBike.setSuspension("Full");  
myBike.setType("Trail");  
myBike.displayBikeDetails();  
}  
}
```

```
Bike Make: Oracle Bikes  
HandleBars: Drop  
Tyres: Off-road  
Seat Type: Comfort  
Suspension: Full  
Type: Trail
```